On Sampling, Anonymization, and Differential Privacy: Or, *k*-Anonymization Meets Differential Privacy

Ninghui Li
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
ninghui@cs.purdue.edu

Wahbeh Qardaji
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
wqardaji@cs.purdue.edu

Dong Su
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
su17@cs.purdue.edu

ABSTRACT

This paper aims at answering the following two questions in privacy-preserving data analysis and publishing: What formal privacy guarantee (if any) does k-anonymization provide? How to benefit from the adversary's uncertainty about the data? We have found that random sampling provides a connection that helps answer these two questions, as sampling can create uncertainty. The main result of the paper is that k-anonymization, when done "safely", and when preceded with a random sampling step, satisfies (ϵ, δ)-differential privacy with reasonable parameters. This result illustrates that "hiding in a crowd of k" indeed offers some privacy guarantees. This result also suggests an alternative approach to output perturbation for satisfying differential privacy: namely, adding a random sampling step in the beginning and pruning results that are too sensitive to change of a single tuple. Regarding the second question, we provide both positive and negative results. On the positive side, we show that adding a random-sampling pre-processing step to a differentially-private algorithm can greatly amplify the level of privacy protection. Hence, when given a dataset resulted from sampling, one can utilize a much large privacy budget. On the negative side, any privacy notion that takes advantage of the adversary's uncertainty likely does not compose. We discuss what these results imply in practice.

1. INTRODUCTION

In this paper we deal with the problem of using data in a privacy-preserving way. We consider the scenario where a trusted curator obtains a dataset by gathering private information from a large number of respondents, and then make usage of the dataset while protecting the privacy of respondents. The curator may learn and release to the public statistical facts about the underlying population. Alternatively, the curator may publish a sanitized (or, "anonymized") version of the dataset so that other parties can use the data to perform any analysis they are interested in.

This paper aims at answering the following two questions in privacy-preserving data analysis and publishing. The first is: What formal privacy guarantee (if any) does k-anonymization methods provide? k-Anonymization methods have been studied extensively in the database community, but have been known to lack strong privacy guarantees. The second question is: How to benefit from the adversary's uncertainty about the data? More specifically, can we come up a meaningful relaxation of differential privacy [8, 9] by exploiting the adversary's uncertainty about the dataset? We now discuss these two motivations in more details.

The k-anonymity notion was introduced by Sweeny and Samarati [30, 29, 27, 28] for privacy-preserving microdata publishing. This notion has been very influential. Many k-anonymization methods have been developed over the last decades; it has also been extensively applied to other problems such as location privacy [14]. The k-anonymity notion requires that when only certain attributes, known as quasi-identifiers (QIDs), are considered, each tuple in a k-anonymized dataset should appear at least k times. In this paper, we consider a version of k-anonymity which treats all attributes as QIDs. We show that even satisfying this strong version of k-anonymity does not protect against re-identification attacks. In addition, we identify the privacy vulnerabilities of existing k-anonymization algorithms. We then define classes of k-anonymization algorithms that are "strongly-safe" and "ϵ-safe", which avoid the privacy vulnerabilities of existing k-anonymization algorithms. The question we aim to answer is whether these safe k-anonymization methods would provide strong enough privacy guarantee in practice.

The notion of differential privacy was introduced by Dwork et al. [8, 11]. An algorithm $\mathcal{A}$ satisfies ϵ-Differential Privacy (ϵ-DP) if and only if for any two neighboring datasets D and D', the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$ differ at most by a multiplicative factor of e^ϵ. A relaxed version of ϵ-DP, which we use (ϵ, δ)-DP to denote, allows an error probability bounded by δ. Satisfying differential privacy ensures that even if the adversary has full knowledge of the values of a tuple t, as well as full knowledge of what other tuples are in the dataset, and is only uncertain about whether t is in the input dataset, the adversary cannot tell whether t is in the dataset or not beyond a certain confidence level. As in most data publishing scenarios, the adversary is unlikely to have precise information about all other tuples in a dataset. It is desirable to exploit this uncertainty to define a relaxed version of differential privacy, which can be easier to satisfy.

We have found that sampling provides the link between our two goals. The main result in this paper is that sampling plus "safe" k-anonymization satisfies (ϵ, δ)-DP. This result leads us to study the relationship between sampling and differential privacy. We say that an algorithm satisfies differential privacy under sampling if the algorithm preceded with a random sampling step satisfies differential privacy.

Results about differential privacy under sampling both are of theoretical interest and have practical relevance. Sampling is a natural way to model the adversary's uncertainty about the data; thus this helps understand how to take advantage of this uncertainty in private data analysis. On the practical side, many data publishing scenarios already involve a random sampling step. Sometimes this sampling step is explicit, when one has a large dataset and wishes to release only a much smaller for research, such as the US census bureau's 1-percent Public Use Microdata Sample. Sometimes, this sampling step is implicit; because the respondents are randomly selected, one can view the dataset as resulted from sampling.

The contributions of this paper are as follows:

- We prove that safe k-anonymization algorithm, when preceded by a random sampling step, provides (ϵ, δ)-differential privacy with reasonable parameters.

  In the literature, k-anonymization and differential privacy have been viewed as very different privacy guarantees: k-anonymization is syntactic, and differential privacy is algorithmic and provides semantic privacy guarantees. Our result is, to our knowledge, the first to link k-anonymization with differential privacy. It illustrates that "hiding in a crowd of k" indeed offers privacy guarantees.

  This result also provides a new way of satisfying differential privacy. Existing techniques for satisfying differential privacy rely on output perturbation, that is, adding noise to the query outputs. Our result suggests an alternative approach. Rather than adding noise to the output, one can add a random sampling step in the beginning and prune results that are too sensitive to changes of individual tuples (i.e., tuples that violate k-anonymity).

- We show both positive and negative results on utilizing the adversary's uncertainty about the data. On the positive side, we show that random sampling has a privacy amplification effect for (ϵ, δ)-DP. For an algorithm that satisfies (ϵ, δ)-DP, adding a sampling step with probability β reduces both $e^{\epsilon} - 1$ and δ by a factor of β. For example, applying an algorithm that achieves ($\ln 2 \approx 0.69$)-differential privacy on dataset sampled with 0.1 probability can achieve overall ($\ln 1.1 \approx 0.095$)-differential privacy.

  On the negative side, we show that any privacy notion that exploits the adversary's uncertainty about the data is unlikely to compose, in the sense that publishing the output from two algorithms together may be non-private.

  Our results suggest the following approaches to take advantage of the fact that the input dataset is resulted from explicit or implicit sampling. If one applies algorithms that satisfy (ϵ, δ)-DP, then one can allow a larger privacy budget because of sampling. If one applies an algorithm that does not satisfy (ϵ, δ)-DP, but satisfies (ϵ, δ)-DP under sampling, then it is safe to apply the algorithm once. However, if one has a large dataset, one can repeated sample and then apply the algorithm on each newly sampled dataset.

The rest of the paper is organized as follows. We study the relationship between differential privacy and sampling in Section 2. We study k-anonymization and prove our main result in Section 3. We discuss related work in Section 4 and conclude in Section 5. An appendix includes proofs not found in the main body.

2. DIFFERENTIAL PRIVACY UNDER SAMPLING

2.1 Differential Privacy

Differential privacy formalizes the following protection objective: if a disclosure occurs when an individual participates in the database, then the same disclosure also occurs with similar probability (within a small multiplicative factor) even when the individual does not participate. More formally, differential privacy requires that, given two input datasets that differ only in one tuple, the output distributions of the algorithm on these two datasets should be close.

DEFINITION 1. ***[ϵ-Differential Privacy [8, 11] (ϵ-DP)]:*** *A randomized algorithm $\mathcal{A}$ gives ϵ-differential privacy if for any pair of neighboring datasets D and D', and any $O \subseteq \mathsf{Range}(\mathcal{A})$,*

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O] \quad (1)$$

Intuitively, ϵ-DP offers strong privacy protection. If $\mathcal{A}$ satisfies ϵ-DP, one can claim that publishing $\mathcal{A}(D)$ does not violate the privacy of any tuple t in D, because even if one leaves t out of the dataset, in which case the privacy of t can be considered to be protected, one may still publish the same outputs with a similar probability.

In practice, ϵ-DP can be too strong to satisfy in some scenarios. A commonly used relaxation is to allow a small error probability δ.

DEFINITION 2. ***[(ϵ, δ)-Differential Privacy [10] ((ϵ, δ)-DP)]:*** *A randomized algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy, if for any pair of neighboring datasets D and D' and for any $O \subseteq \mathsf{Range}(\mathcal{A})$:*

$$Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in O] + \delta$$

Existing methods to satisfy differential privacy includes adding Laplace noise proportional to the query's global sensitivity [8, 11], adding noise related to the smooth bound of the query's local sensitivity [26], and the exponential mechanism to select a result among all possible results [25].

2.2 Uncertain Background Knowledge

One of our goals is to develop a further relaxation of differential privacy that can be more easily satisfied. The intuition that we wanted to exploit is the adversary's uncertainty about the underlying dataset. The (ϵ, δ)-DP notion ensures that when an adversary is uncertain about whether one tuple t is present in the input dataset, even when the adversary knows the *precise information* all other tuples in the input dataset, the adversary cannot tell based on the output whether t is in the input or not. We believe that it is reasonable to relax the assumption to that the adversary knows all attributes of a tuple t (but not whether t is in the dataset), and in addition statistical information about the rest of the dataset D. The privacy notion should prevent such an adversary from substantially distinguishing between D and $D \cup \{t\}$ based on the output.

The desire to exploit adversary's uncertainty is shared by other researchers. For example, Adam Smith's blog post summarizing the Workshop on Statistical and Learning-Theoretic Challenges in Data Privacy includes a section on relaxed definitions of privacy with meaningful semantics: "it would be nice to see meaningful definitions of privacy in statistical databases that exploit the adversary's uncertainty about the data. The normal approach to this is to specify a set of allowable prior distributions on the data (from the adversary's point of view). However, one has to be careful. The versions I have seen are quite brittle."[1]

9 [1] *http://adamdsmith.wordpress.com/2010/03/04/ipam-workshop-wrap-up/*

Some degree of brittleness may be unavoidable. It appears that any privacy notion that takes advantage of the adversary's uncertainty about the data is not robust under composition, which requires that given two algorithms that both satisfy the privacy notion, their composition, i.e., applying both algorithms to the same input dataset and then publish both outputs, also satisfies the privacy notion.

Consider the following two algorithms. Let $r(D)$ be the predicate that D contains an odd number of tuples, and $s(D)$ be a sensitive predicate, e.g., whether a tuple t is in D. Algorithm $\mathcal{A}_1(D)$ outputs $r(D)$, and $\mathcal{A}_2(D)$ outputs $r(D)$ XOR $s(D)$. Both $\mathcal{A}_1$ and $\mathcal{A}_2$ should satisfy a privacy notion that assumes that the adversary is uncertain about the data, because there is no reason that the adversary should know the exact number of the tuples. However, the composition of $\mathcal{A}_1$ and $\mathcal{A}_2$ leaks $r(D)$. More generally, for any privacy definition that exploits the adversary's uncertainty about data, there exists at least one predicate that the adversary is uncertain about. Then one algorithm can output that predicate, and a second algorithm can output that predicate XOR's with a predicate that results in privacy leakage; and they does not compose.

The above observation suggests that no such definition should be used in the interactive setting of answering multiple queries. If, however, one intends to publish a dataset in the non-interactive setting only once, then the inability to compose may be an acceptable limitation.

2.3 Differential Privacy under Sampling

One natural approach to capturing the adversary's uncertainty about the input data is to add a sampling step. We introduce the following definition, called $(\beta, \epsilon, \delta)$-Differential Privacy under Sampling ($(\beta, \epsilon, \delta)$-DPS for short).

DEFINITION 3 (DIFFERENTIAL PRIVACY UNDER SAMPLING). *An algorithm $\mathcal{A}$ gives $(\beta, \epsilon, \delta)$-DPS if and only if $\beta > \delta$ and the algorithm $\mathcal{A}^\beta$ gives (ϵ, δ)-DP, where $\mathcal{A}^\beta$ denotes the algorithm to first sample with probability β (include each tuple in the input dataset with probability β), and then apply $\mathcal{A}$ to the sampled dataset.*

The above definition requires $\beta > \delta$ because any algorithm trivially satisfies $(\beta, 0, \delta)$-DPS when $\beta \leq \delta$. This is because when two datasets differ only by one tuple, sampling from them with the probability β will result in exactly the same output with probability $1 - \beta$. However, when $\beta \gg \delta$, the notion of $(\beta, \epsilon, \delta)$-DPS is both nontrivial to satisfy and a nontrivial relaxation of (ϵ, δ)-DP, as shown by our results in Section 3. There we show that existing k-anonymization algorithms do not satisfy $(\beta, \epsilon, \delta)$-DPS, and have privacy vulnerabilities, and that safe (and possibly deterministic) k-anonymization satisfies $(\beta, \epsilon, \delta)$-DPS, while violating (ϵ, δ)-DP for any $\delta < 1$.

2.4 The Amplification Effect of Sampling

An interesting feature of the $(\beta, \epsilon, \delta)$-DPS notion is that there is a connection between the privacy parameters ϵ, δ and the sampling rate β. The following theorem shows that by employing a smaller sampling rate, one can achieve a stronger privacy protection (i.e., smaller values for ϵ and δ).

THEOREM 1. *Any algorithm that satisfies $(\beta_1, \epsilon_1, \delta_1)$-DPS also satisfies $(\beta_2, \epsilon_2, \delta_2)$-DPS for any $\beta_2 < \beta_1$, where $\epsilon_2 = \ln\left(1 + \left(\frac{\beta_2}{\beta_1}(e^{\epsilon_1} - 1)\right)\right)$, and $\delta_2 = \frac{\beta_2}{\beta_1}\delta_1$.*

See Appendix A.1 for the proof.

β	e^ϵ	ϵ	δ
1	11	$\ln 11 \approx 2.40$	10^{-5}
0.1	2	$\ln 2 \approx 0.69$	10^{-6}
0.01	1.1	$\ln 1.1 \approx 0.095$	10^{-7}

β	ϵ
1	1
0.1	0.159
0.01	0.017

Table 1: Effect of privacy parameters under sampling.

An equivalent way to write $\epsilon_2 = \ln\left(1 + \left(\frac{\beta_2}{\beta_1}(e^{\epsilon_1} - 1)\right)\right)$ is

$$\frac{e^{\epsilon_2} - 1}{e^{\epsilon_1} - 1} = \frac{\beta_2}{\beta_1}.$$

In other words, by decreasing the sampling probability, one obtains proportional decreases in $e^\epsilon - 1$ and δ, improving the privacy protection. Hence, when one possesses a randomly sampled dataset, then one can use much relaxed privacy budget ϵ and error toleration δ. To see the effects of this, in Table 1 we show the privacy parameters for an algorithm that satisfies $(\ln 11, 10^{-5})$-DP, and an algorithm that satisfies $(1, 0)$-DP under sampling rate 0.1 and 0.01.

Smith's blog [2] includes an "amplification" lemma for differential privacy, which was used implicitly in the design of a PAC learner for the parity class in [17]. The lemma states that an algorithm that satisfies $(\epsilon = 1)$-DP, when preceded by random sampling with rate β, satisfies (2β)-DP. Theorem 1 exploits similar observations, but is more general in that it applies to (ϵ, δ)-DP, rather than ϵ-DP, and that it also applies to arbitrary values of ϵ. Our result is also slightly tighter; for example, for the special case of $\epsilon = 1$ and $\beta = 0.1$, we give a result of 0.159 as opposed to $2\beta = 0.2$.

2.5 Properties of $(\beta, \epsilon, \delta)$-DPS

While the $(\beta, \epsilon, \delta)$-DPS notion does not compose. It does have several other desirable properties. In [19], Kifer and Lin identified two privacy axioms when they defined the generic differential privacy. The *Transformation Invariance* axiom states that given an algorithm $\mathcal{A}$ that satisfies a privacy notion, adding any post-processing step operating on $\mathcal{A}$'s output should still satisfy the privacy notion. The *Privacy Axiom of Choice* axiom states that given two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ that both satisfy a privacy notion, then a new algorithm that chooses $\mathcal{A}_1$ with probability p and $\mathcal{A}_2$ with probability $1 - p$ should also satisfy the notion. We now show that $(\beta, \epsilon, \delta)$-DPS satisfies both axioms.

THEOREM 2. *Given $\mathcal{A}_1$ that satisfies $(\beta, \epsilon, \delta)$-DPS and any algorithm $\mathcal{A}_2$, $\mathcal{A}(D) = \mathcal{A}_2(\mathcal{A}_1(D))$ satisfies $(\beta, \epsilon, \delta)$-DPS.*

PROOF. Assume, for the sake of contradiction, that $\mathcal{A}(D)$ does not satisfy $(\beta, \epsilon, \delta)$-DPS, then there exist neighboring D and D' and $O \subseteq \mathsf{Range}(\mathcal{A}_2)$ such that

$$\Pr[\mathcal{A}_2(\mathcal{A}_1^\beta(D)) \in O] > e^\epsilon \Pr[\mathcal{A}_2(\mathcal{A}_1^\beta(D')) \in O] + \delta$$

Consider all S's in $\mathsf{Range}(\mathcal{A}_1)$, let $q(S) = \Pr[\mathcal{A}_2(S) \in O]$, and let $p(S) = \Pr[\mathcal{A}_1^\beta(D) = S]$ and $p'(S) = \Pr[\mathcal{A}_1^\beta(D') = S]$. Then we have

$$\sum_{S \in \mathsf{Range}(\mathcal{A}_1)} p(S)q(S) > e^\epsilon \sum_{S \in \mathsf{Range}(\mathcal{A}_1)} p'(S)q(S) + \delta.$$

We partition $\mathsf{Range}(\mathcal{A}_1)$ into $\mathcal{S}_1 = \{S \mid p(S) > e^\epsilon p'(S)\}$ and $\mathcal{S}_2 = \{S \mid p(S) \leq e^\epsilon p'(S)\}$. Rewriting the above inequality, we

9[2] *http://adamdsmith.wordpress.com/2009/09/02/sample-secrecy/*

have

$$\begin{array}{rl} & \sum_{S\in\mathcal{S}_1} p(S)q(S) + \sum_{S\in\mathcal{S}_2} p(S)q(S) \\ > & e^{\epsilon}\sum_{S\in\mathcal{S}_1} p'(S)q(S) + e^{\epsilon}\sum_{S\in\mathcal{S}_2} p'(S)q(S) + \delta \end{array}$$

Consider the sum over $\mathcal{S}_2$, we have

$$\sum_{S\in\mathcal{S}_2} p(S)q(S) \le e^{\epsilon}\sum_{S\in\mathcal{S}_2} p'(S)q(S)$$

Subtracting the above from previous, we have

$$\sum_{S\in\mathcal{S}_1} p(S)q(S) > e^{\epsilon}\sum_{S\in\mathcal{S}_1} p'(S)q(S) + \delta$$

For each $S \in \mathcal{S}_1$, we have $p(S)(1-q(S)) > e^{\epsilon}p'(S)(1-q(S))$, and thus

$$\sum_{S\in\mathcal{S}_1} p(S)(1-q(S)) > e^{\epsilon}\sum_{S\in\mathcal{S}_1} p'(S)(1-q(S))$$

Summing up the above two inequalities, we have

$$\sum_{S\in\mathcal{S}_1} p(S) > e^{\epsilon}\sum_{S\in\mathcal{S}_1} p'(S) + \delta$$

This contradicts that $\mathcal{A}_1$ satisfies (β,ϵ,δ)-DPS. □

THEOREM 3. *Given two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ that both satisfy (β,ϵ,δ)-DPS, for any $p \in [0,1]$, let $\mathcal{A}_p(D)$ be the algorithm that outputs $\mathcal{A}_1(D)$ with probability p and $\mathcal{A}_2(D)$ with probability $1-p$, then $\mathcal{A}_p$ satisfies (β,ϵ,δ)-DPS.*

PROOF. Since both $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy (β,ϵ,δ)-DPS, for any pair of neighboring datasets D and D' and for any $O \in \mathsf{Range}(\mathcal{A}_1) \cup \mathsf{Range}(\mathcal{A}_2)$, we have

$$\begin{array}{rl} & \Pr[\mathcal{A}_p(D) \in O] \\ = & p\Pr[\mathcal{A}_1(D)\in O] + (1-p)\Pr[\mathcal{A}_2(D)\in O] \\ \le & p(e^{\epsilon}\Pr[\mathcal{A}_1(D')\in O] + \delta) + (1-p)(e^{\epsilon}\Pr[\mathcal{A}_2(D')\in O] + \delta) \\ = & e^{\epsilon}(p\Pr[\mathcal{A}_1(D')\in O] + (1-p)\Pr[\mathcal{A}_2(D')\in O]) + \delta \\ = & e^{\epsilon}\Pr[\mathcal{A}_p(D')\in O] + \delta. \end{array}$$

Therefore, the algorithm $\mathcal{A}_p$ also satisfies (β,ϵ,δ)-DPS. □

2.6 More Non-Composability

From observations in Section 2.2, we expect that (β,ϵ,δ)-DPS does not compose. However, one would expect that combining an algorithm that satisfies $(\beta,\epsilon_1,\delta)$-DPS and one that satisfies ϵ_2-DP should result in an algorithm that satisfies the weaker (β,ϵ,δ)-DPS, where ϵ is some function of ϵ_1 and ϵ_2. Such a weaker form of composability is useful in that given a dataset that is resulted from random sampling, one can publish it in a way that satisfies (β,ϵ,δ)-DPS, while at the same time answering queries using mechanisms that satisfy ϵ-DP. Surprisingly, even such a weak form of composability does not hold.

Consider the following two algorithms operating on datasets in which each tuple has two fields: gender and name. Let $r(D)$ be the predicate that D contains more male than female, and $s(D)$ be a sensitive predicate, such as whether D contains a specific tuple. The algorithm $\mathcal{A}_1(D)$ outputs $r(D)$ XOR $s(D)$ when D contains a sufficient number of tuples (say, 1000), and outputs false otherwise. And $\mathcal{A}_2(D)$ outputs the percentage of tuples in D that are male with Laplacian noise [11].

Clearly $\mathcal{A}_2(D)$ satisfies ϵ-DP. $\mathcal{A}_1$ satisfies (β,ϵ,δ)-DPS for any β that is not too close to 1. Let T and T' be the random variables resulted from sampling from D and D' respectively. Only when the dataset size is large enough, would $\mathcal{A}_1$ output information that depends on the input data. When D and D' contain a large number of tuples and differ only by one, $r(T)$ and $r(T')$ have essentially the same distribution, taking the value true with probability very close to 0.5, making $\mathcal{A}_1(T)$ and $\mathcal{A}_1(T')$ having a similar distribution. Combining $\mathcal{A}_1$ and $\mathcal{A}_2$, however, is non-private. Using $\mathcal{A}_2(T)$ one obtains a highly accurate estimate of the predicate $r(T)$, enabling the adversary to learn $s(T)$ with high probability.

More specifically, let D and D' be two datasets such that $s(D)$ is false, $s(D')$ is true (i.e., D' contains the tuple we are checking), and they each contain 10,000 tuples, half male and half female. Consider sampling probability $\beta = 0.5$, and the event that $\mathcal{A}_1$ outputs false, and $\mathcal{A}_2$ outputs $p \ge 0.5$. Let T and T' be the random variables resulted from sampling from D and D' respectively, then we have

$$\begin{array}{rl} \Pr[s(T) = \text{true}] & = 0 \\ \Pr[s(T') = \text{true}] & = 1/2 \\ \Pr[r(T) = \text{true} \mid \mathcal{A}_2(T) \ge 0.5] & \approx 1, \\ \Pr[r(T') = \text{true} \mid \mathcal{A}_2(T') \ge 0.5] & \approx 1 \end{array}$$

and

$$\begin{array}{rl} & \Pr[\mathcal{A}_2(T) \ge 0.5 \wedge \mathcal{A}_1(T) = \text{false}] \\ = & \Pr[\mathcal{A}_2(T) \ge 0.5]\Pr[r(T) = s(T) \mid \mathcal{A}_2(T) \ge 0.5] \\ \approx & \Pr[\mathcal{A}_2(T) \ge 0.5]\Pr[s(T) = \text{true}] \\ = & 0, \end{array}$$

while

$$\begin{array}{rl} & \Pr[\mathcal{A}_2(T') \ge 0.5 \wedge \mathcal{A}_1(T') = \text{false}] \\ \approx & \Pr[\mathcal{A}_2(T') \ge 0.5]\Pr[s(T') = \text{true} \mid \mathcal{A}_2(T') \ge 0.5] \\ = & \Pr[\mathcal{A}_2(T') \ge 0.5]\Pr[s(T') = \text{true}] \\ \approx & 1/4. \end{array}$$

This result is somewhat surprising. After all, any mechanism that satisfies ϵ-DP should not be leaking private information about the underlying datasets. How could adding a differentially private mechanism destroys the privacy protection of another mechanism? Our understanding is that satisfying (β,ϵ,δ)-DPS can be achieved by relying on the adversary's uncertainty. The adversary knows only that the dataset is from a large set of candidates. While ϵ-DP ensures that adjacent datasets are difficult to distinguish, these candidates are not all adjacent and can indeed be quite far apart. Hence obtaining one ϵ-DP answer may dramatically change the probability of which candidates are possible, removing some degree of uncertainty, destroying any privacy protection that relies on exactly that uncertainty.

This inability for a (β,ϵ,δ)-DPS mechanism to compose with a ϵ-DP mechanism suggests that (β,ϵ,δ)-DPS mechanisms should be applied alone. Hence they are not suitable for the interactive mode, but only suitable for the non-interactive mode of data publishing. Furthermore, it also suggests that mechanisms satisfying ϵ-DP should be used carefully as well, as its output may break other mechanisms' (albeit weaker) privacy guarantees.

2.7 Benefiting from Sampling

We observe that in many data publishing scenarios, random sampling is an inherent step. For example, the census bureau publishes a 1-percent microdata sample. In many research settings (such as when Netflix wants to publishing movie ratings), it is sufficient to publish a random sample of the dataset. Many times, even when the dataset is not the result of explicit sampling, one can view it as result of implicit sampling, because the process of selecting respondents involves randomness.

The natural question is how one can benefit from such explicit or implicit sampling. Our results provide the following answers. The first way is to limit oneself to mechanisms that satisfy

(ϵ, δ)-DP, then the uncertainty resulted from sampling enables one to use a larger privacy budget because of the amplification result in Theorem 1. The second way is to use a mechanism that does not satisfy (ϵ, δ)-DP, but satisfies $(\beta, \epsilon, \delta)$-DPS, such as safe k-anonymization, which we will study in Section 3. However, this way of benefiting from sampling can be enjoyed only once; one cannot use the same dataset to answer other queries, even when using mechanisms that satisfy ϵ-DP.

There, however, does exist a more flexible way to use a mechanism that satisfies only $(\beta, \epsilon, \delta)$-DPS. When one has a large dataset, one can sample a dataset, apply the mechanism, publish the result, and discard the intermediate sampled dataset. Because of the composability of (ϵ, δ)-DP, this approach can be applied multiple times so long as each time one performs a fresh sampling. One can also use multiple mechanisms that satisfy (ϵ, δ)-DP on a newly sampled dataset.

We point out that the benefit of sampling should not be viewed as just "throwing away data"; sampling's main benefit is to introduce uncertainty. Given a dataset, one could sample with, say, $\beta = 0.2$ for many (say, 50) times, and apply a mechanism that satisfies $(0.2, 0.02, 0)$-DPS to each sampled dataset and publish the results. With high probability, each tuple is included in at least one of the sampled datasets. That is, in some sense, no tuple is thrown away. However, as each sampling and publishing satisfies (ϵ, δ)-DP, and (ϵ, δ)-DP composes, publishing the 50 outputs still satisfies (ϵ, δ)-DP for $\epsilon = 1, \delta = 0$.

In summary, sampling creates uncertainty for the adversary. While the benefit due to this uncertainty is easy to lose because the uncertainty can be jeopardized by answering any query on it, this uncertainty is also easy to gain, as each sampling introduces fresh uncertainty.

3. SAFE K-ANONYMIZATION MEETS DIFFERENTIAL PRIVACY

In this section we show that k-anonymization, when performed in a "safe" way, satisfies $(\beta, \epsilon, \delta)$-DPS. That is, safe k-anonymization, when preceded by a random sampling step, satisfies (ϵ, δ)-differential privacy.

3.1 An Analysis of k-Anonymity

The development of k-anonymity was motivated by a well publicized privacy incident [30]. The Group Insurance Commission (GIC) published a supposedly anonymized dataset recording the medical visits of patients managed under its insurance plan. While the obvious personal identifiers (such as name and address) were removed, the published data included zip code, date of birth, and gender, which are sufficient to uniquely identify a significant fraction of the population. Sweeney [30] showed that by correlating this data with the publicly available Voter Registration List for Cambridge Massachusetts, medical visits for many individuals can be easily identified, including those of William Weld, a former governor of Massachusetts. We note that even without access to the public voter registration list, the same privacy breaches can occur. Many individuals' birthdate, gender and zip code are public information. This is especially the case with the advent of social media, including Facebook, where users share seemingly innocuous personal information to the public. The GIC re-identification attack directly motivated the development of the k-anonymity privacy notion.

DEFINITION 4. ***[k-Anonymity, the privacy notion] [30]:*** *A published table satisfies k-anonymity relative to a set of QID attributes if and only if when the table is projected to include only the QIDs, every tuple appears at least k times.*

Quasi-identifiers vs. Sensitive Attributes? A first problem with Definition 4 is that it requires the division of all attributes into quasi-identifiers (QIDs) and sensitive attributes (SA), where the adversary is assumed to know the QIDs, but not SAs. This separation, however, is very hard to obtain in practice. Even though only some attributes are used in the GIC incident, it is difficult to assume that they are the only QIDs. Other attributes in the GIC data include visit date, diagnosis, etc. There may well exist an adversary who knows this information about some individuals, and if with this knowledge these individuals' record can be re-identified, it is still a serious privacy breach.

The same difficulty is true for publishing any kind of census, medical, or transactional data. When publishing anonymized microdata, one has to defend against all kinds of adversaries, some know one set of attributes, and others know different sets. An attribute about one individual may be known by some adversaries, and unknown (and should be considered sensitive) for other adversaries.

Any separation between QIDs and SAs is essentially making assumptions about the adversary's background knowledge that can be easily violated, rendering any privacy protection invalid. Hence we consider a strengthened version of k-anonymity by treating all attributes as QIDs. This is stronger than using any subset of attributes as QIDs. This strengthened version of k-anonymity avoids making assumption about the adversary's background knowledge about which attributes are known and what are not. This has been used in the context of anonymizing transaction data [16].

Weakness of the k-Anonymity Notion. With the strengthened version of k-anonymity, one might expect that it should stop re-identification attacks. To satisfy this notion, each tuple in the output is blended in a group of at least k tuples that are the same. This follows the appealing principle that "privacy means hiding in a crowd". The intuition is that as there are at least $k-1$ other tuples that look exactly the same, one cannot re-identify which tuple in the output corresponds to an individual with probability over $1/k$. Unfortunately, this intuition turns out to be wrong. Only making the syntactic requirement that each tuple appears at least k times does not protect privacy, as a trivial way to satisfy this is to select some tuples from the input and then duplicate each of them k times.

Several other privacy notions have been introduced on the motivation that k-anonymity is not strong enough. Among these are ℓ-diversity [23] and t-closeness [22]. In these approaches, it is observed that even if k-anonymity is achieved, information about sensitive attributes can still be learned, perhaps due to the uneven distribution of their values. This line of work, however, still requires the problematic assumption that there is a separation between QIDs and SAs, and that the adversary knows only the QIDs. In other words, while they correctly assert that k-anonymity is not strong enough, these definitions did not fix it in the right way.

k-Anonymity vs. k-Anonymization Algorithms. Here we would like to make a clear distinction between the *k-anonymity*, the privacy notion, and *k-anonymization algorithms*.

Many k-anonymization algorithms have been developed in the literature. Given input datasets, they aim at producing anonymized versions of the input datasets that satisfy k-anonymity. That the k-anonymity privacy notion is weak means that producing outputs that *satisfying k-anonymity alone* is insufficient for privacy protection. However, this does not automatically mean that all k-anonymization have privacy vulnerabilities. We now show that the algorithms that have been developed in the literature are indeed vulnerable to re-identification attacks. Consider the following anonymization scheme, which represents several proposed algorithms for k-anonymity [5, 21].

ALGORITHM 1. ***[Clustering and Local Recoding (CLR)]:*** *First, group input tuples into clusters such that each cluster has at least k tuples. For example, one method of grouping is the Mondrian algorithm [21]. One could also use some clustering method based on some distance measurement (e.g., [5]). Then, for each tuple, replace each attribute value with a generalized value that represents all values for that attribute in the cluster.*

CLR algorithms are vulnerable when some tuples contain extreme values. Even if the output satisfies k-anonymity, the generalized value depends on the extreme values of some tuples; hence from the output an adversary can infer that one's tuple is in the dataset and can thus infer these values. For example, suppose the dataset records the net worth of some individuals in a town. Further suppose that it is known that only one individual in the town has net worth over $10 million. When given a $(k = 20)$-anonymized output dataset containing one group of tuples that all have $[900K, 35M]$ as the generalized net worth value, what can one conclude? At least the following: the rich individual is in the dataset; the individual's tuple is in the group; and the individual's net worth is $35 million. It would be difficult to say that because in the output dataset, there are at least 19 other tuples that are exactly the same, then the individual cannot be re-identified with probability $1/20$.

Similar weaknesses exist for other k-anonymization algorithm in the literature, for example, those computing a generalization scheme based on the input dataset [16]. With all these algorithms, the presence and non-presence of some extreme values will affect the resulted generalization scheme, leaking information.

As these algorithms are sensitive to the presence of a single tuple with extreme values, they do not satisfy $(\beta, \epsilon, \delta)$-DPS when $\beta > \delta$, since sampling with β will result the presence of the tuple selected with probability β.

3.2 Towards "Safe" k-Anonymization

We have shown that k-anonymity (even when all attributes are treated as QID) does not provide adequate protection, nor do existing k-anonymization algorithms. One natural question is: Is this because the intuition "hidden in a crowd" fails to provide privacy protection, or is it because the definition of k-anonymity fails to correctly capture "hidden in a crowd"?

We believe that the answer is the latter. The notion of k-anonymity implicitly assumes that there is a one-to-one relation g between the input tuples and the output tuples, i.e., given input D, the output dataset is $\{g(t) \mid t \in D\}$. When there are k output tuples that are the same, there must exist k input tuples that are indistinguishable based only on their corresponding outputs. However, this relation g itself can be overly dependent on one or a few input tuples. For instance, consider the example above with the extreme value. Choosing $[900K, 35M]$ as the generalized value depends on the single input tuple with value $35M$; hence all tuples that contain this generalized value are directly affected by one tuple's presence, and the tuple is not really "hiding in a crowd".

An intriguing question is: If a k-anonymization algorithm uses a mapping that does not overly depend on any individual tuple, does such an algorithm provide an adequate level of privacy protection? To answer this question, we first formalize such algorithms as safe k-Anonymization algorithms.

Intuitively, an k-anonymization algorithm $\mathcal{A}$ takes as input a dataset D and a value k and produces an output dataset $S = \mathcal{A}(D)$. In order to define "safe" anonymization algorithms, we require each anonymization algorithm $\mathcal{A}$ to be specified in two steps. The first step, $\mathcal{A}_m$, outputs a mapping function $g : D \rightarrow T$, where T is the set of all possible tuples. The second step applies g to all tuples in D. That is, $\mathcal{A}(D, k) = \mathsf{Apply}(\mathcal{A}_m(D, k), D, k)$, where Apply is defined as follows.

```
Apply($g, D, k$)
  $S \leftarrow \emptyset$
  for all $t \in D$ do
    $S \leftarrow S \cup g(t)$
  end for
  for all $s \in S$ do
    if $s$ appears less than $k$ times in $S$ then
      remove all occurrences of $s$ from $S$
    end if
  end for
  return $S$
```

We note that all existing k-anonymization algorithms can be modeled this way, as there is no limitation on the the form of $\mathcal{A}_m$'s output g. In the extreme case, g can be described as a table matching each tuple in D to the desired output tuple.

DEFINITION 5 (STRONGLY-SAFE ANONYMIZATION). *We say that a k-anonymization algorithm $\mathcal{A}$ is strongly safe if and only if the function $\mathcal{A}_m(D, k)$ is remains constant when D changes, i.e., the mapping g does not depend on its input dataset.*

An example of a strongly-safe k-anonymization algorithm is to always use the same global recoding scheme no matter what dataset is the input.

Intuitively a strongly-safe k-anonymization algorithm provides some level of privacy protection, and the level of privacy protection increases with larger values of k. If any individual's tuple is published, there must exist at least $k - 1$ other tuples in the *input* database that are the same under the recoding scheme; furthermore, the recoding scheme does not depend on the dataset, and one sees only the results of the recoding. Hence in this input dataset, the individual is hidden in a crowd of at least k. However, the following proposition shows that strongly safe k-anonymization algorithms do not satisfy (ϵ, δ)-DP.

PROPOSITION 4. *No strongly-safe k-anonymization algorithm satisfies* (ϵ, δ)-DP *for any* $\delta < 1$.

PROOF. Given a strongly-safe algorithm $\mathcal{A}$, let g be the mapping $\mathcal{A}$ uses. Choose D and D' that differ in one tuple t and D contains $n > k$ tuples t' such that $g(t') = g(t)$. The dataset D' contains $n-1$ copies of such t'. Then, $\mathcal{A}(D)$ and $\mathcal{A}(D')$ contain different numbers of $g(t)$. Let $S = \mathcal{A}(D)$, we have $\Pr[\mathcal{A}(D) = S] = 1$ and $\Pr[\mathcal{A}(D') = S] = 0$. □

3.3 Privacy of Strongly-Safe k-Anonymization

We now show that strongly-safe k-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-differential privacy for a small δ with reasonable values of k and β. We use $f(j; n, \beta)$ to denote the probability mass function for the binomial distribution; that is, $f(j; n, \beta)$ gives the probability of getting exactly j successes in n trials where each trial succeeds with probability β. And we use $F(j; n, \beta)$ to denote the cumulative probability mass function; that is, $F(j; n, \beta) = \sum_{i=0}^{j} f(i; n, \beta)$.

THEOREM 5. *Any strongly-safe k-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-DPS for any $0 < \beta < 1$, $\epsilon \geq -\ln(1-\beta)$, and $\delta = d(k, \beta, \epsilon)$, where the function d is defined as*

$$d(k, \beta, \epsilon) \quad = \quad \max_{n: n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta),$$

where $\gamma = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon}$.

See Appendix A.2 for the proof.

The function d relates the four parameters $\epsilon, \beta, k, \delta$ by requiring $\delta = d(k, \beta, \epsilon)$. Note that the other requirement is that $\epsilon \geq -\ln(1-\beta)$. Among the four parameters, ϵ and δ define the level of privacy protection, while k and β affect the quality of anonymized data. We now examine the relationships among these four parameters.

To compute this, we want to find $n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil$ that maximizes $\sum_{j > \gamma n}^{n} f(j; n, \beta)$. We first observe that $\gamma > \beta$ because

$$\gamma - \beta = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon} - \beta = \frac{(e^\epsilon - 1)(1-\beta)}{e^\epsilon} > 0$$

That is, $\sum_{j > \gamma n}^{n} f(j; n, \beta)$ sums up the tail binomial distribution probabilities for the portion of the tail beyond γn, as shown in Figure 1. Following the intuition behind the law of large numbers, the larger the value of n, the smaller this tail probability. Hence intuitively, choosing the smallest value of n, i.e., $n = n_m = \left\lceil \frac{k}{\gamma} - 1 \right\rceil$ should maximize the formula. Unfortunately, due to the discrete nature of the binomial distribution, the maximum value may not be reached at n_m, but instead at one of the next few local maximal points $\left\lceil \frac{k+1}{\gamma} - 1 \right\rceil$, $\left\lceil \frac{k+2}{\gamma} - 1 \right\rceil$, $\cdots$. Thus we are unable to further simplify the representation of the function $d(k, \beta, \epsilon)$.

We now report the relationships among $\epsilon, \beta, k, \delta$ using numerical computation. In Table 2, we fix $k = 20$ and report the values of δ under different ϵ and β values. The table shows that the values of δ can be very small. We note that with fixed k and β, δ decreases as ϵ increases, which states that the error probability gets smaller when one relaxes the ϵ-bound on the probability ratio. In other words, the more serious a privacy breach, the more unlikely it occurs. The table also shows that with fixed k and ϵ, δ decreases as β decreases, meaning that a smaller sampling probability improves the privacy protection.

In Figure 2, we show the results from examining the relationship between ϵ and δ when we vary $k \in \{5, 10, 20, 30, 50\}$ under fixed $\beta = 0.2$. We plot $\frac{1}{\delta}$ against ϵ for values of $\epsilon > -\ln(1-\beta)$. The figure indicates a negative correlation between ϵ and δ. Furthermore, increasing k has a close to exponential effect of improving privacy protection. For example, when $\epsilon = 2$, increasing k by 10 roughly decreases δ by 10^{-5}.

In Figure 3, we show the results from examining the effect of varying $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$ under a fixed value of $k = 20$. This shows that decreasing β also dramatically improve the privacy protection. The two figures indicate the intricate relationship between privacy and utility.

In Figure 4, we explore this phenomenon that increasing k and decreasing β both improve privacy protection. Starting from $(k = 15, \beta = 0.05)$, each time we double β and find a value k that gives a similar level of privacy protection. We finds that k increases from 15 to 22 (for $\beta = 0.1$), 35 (for $\beta = 0.2$), and 60 (for $\beta = 0.4$).

In Figure 5, we examine the quality of privacy protection for very small k's (from 1 to 5). We choose a very small sampling probability of $\beta = 0.025$. Not surprisingly, when $k = 1$, the privacy protection is entirely from the sampling effect, as the obtained δ value is less than β. However, when $k \geq 2$, we start seeing privacy protection effect from k-anonymization, with δ (< 0.001) significantly smaller than $\beta = 0.025$ when $\epsilon = 2$.

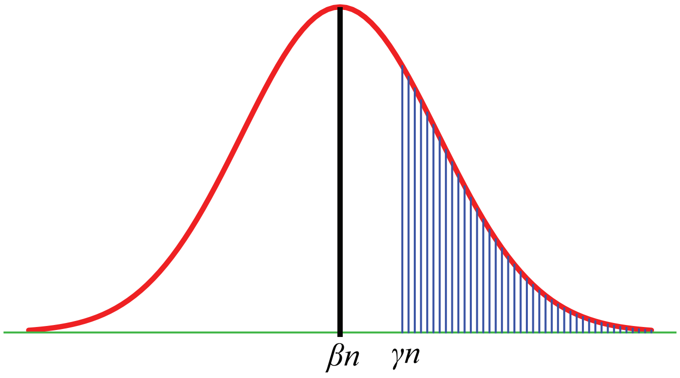

Figure 1: A graph showing the relationship between βn and γn on a binomial curve

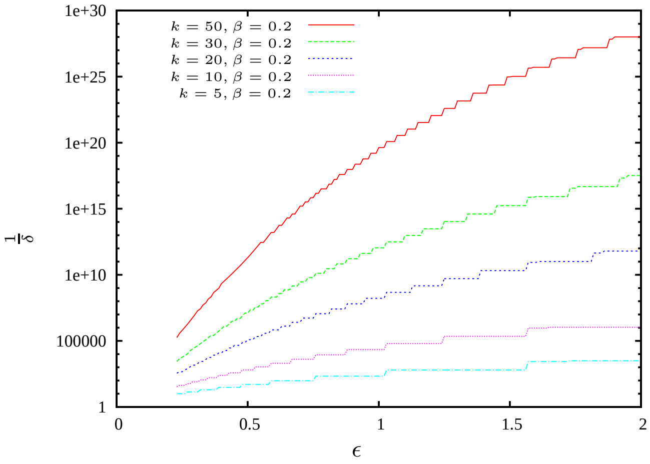

Figure 2: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ if we vary the values of k under fixed β

Finally, in Figure 6 we show the relationship between the privacy parameter ϵ and the utility parameter k if we set the requirement that $\delta \leq 10^{-6}$. The figure shows that smaller values of ϵ can be satisfied for larger values of k. Furthermore, the effect of β over ϵ is quite substantial.

3.4 ϵ-Safe k-Anonymization

In practice, requiring a k-anonymization algorithm to be strongly safe is likely to result in outputs that have low utility. We now relax this requirement to allow the generalization scheme to be chosen in a way that depends on the input dataset, but does not overly depend on any individual tuple.

DEFINITION 6 (ϵ-SAFE k-ANONYMIZATION). *We say that a k-anonymization algorithm $\mathcal{A}$ is ϵ-safe if and only if the function $\mathcal{A}_m$ satisfies ϵ-DP.*

One possible approach to do this is to consider various possible generalization schemes, uses a quality function to assign a quality to each of them, and then uses the exponential mechanism [25] to select in a differentially private way a generalization scheme that gives good utility.

		ϵ					
		0.25	0.5	0.75	1.0	1.5	2.0
β	0.05	6.83×10^{-10}	2.50×10^{-14}	3.19×10^{-17}	1.76×10^{-19}	3.97×10^{-22}	2.00×10^{-24}
	0.1	4.19×10^{-06}	1.61×10^{-09}	3.44×10^{-12}	4.07×10^{-14}	3.22×10^{-16}	1.89×10^{-18}
	0.2	2.16×10^{-03}	8.02×10^{-06}	1.89×10^{-07}	6.03×10^{-09}	4.79×10^{-11}	1.59×10^{-12}

Table 2: A table showing the relationship between β and ϵ in determining the value of δ when k is fixed. In the above $k = 20$, and each cell in the table reports the value of δ under the given values of β and ϵ

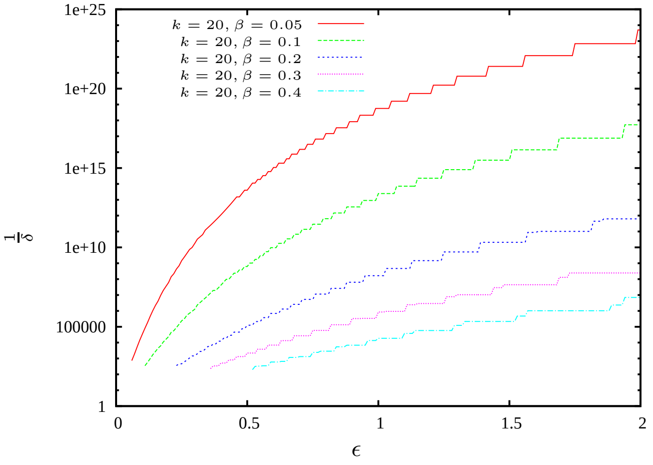

Figure 3: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ if we vary the values of β under fixed k.

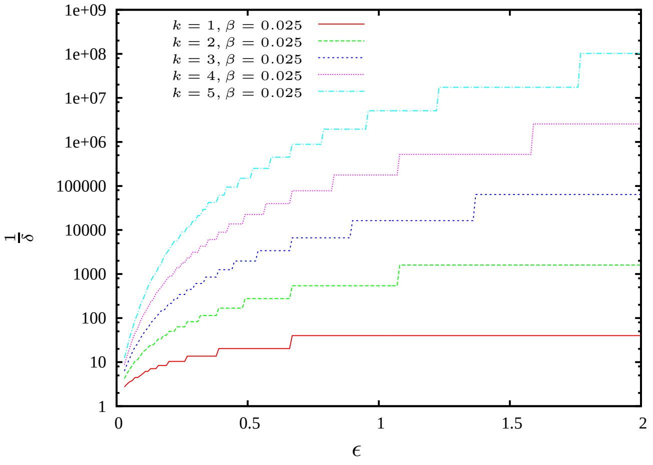

Figure 5: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ with small k's, varying k and fixing β.

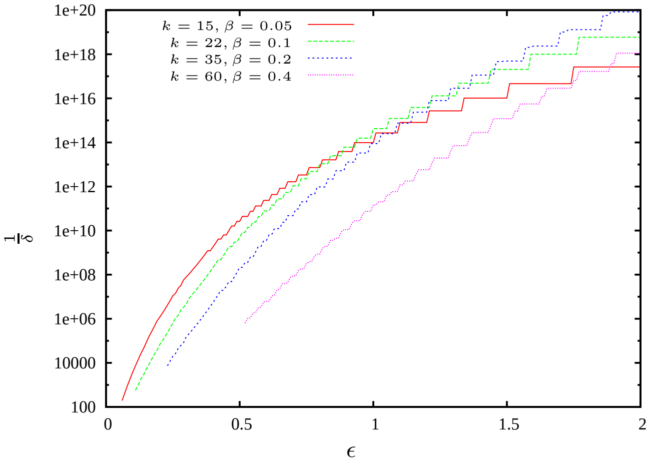

Figure 4: A graph showing the relationship between the values of k needed to achieve roughly the same δ if we double β.

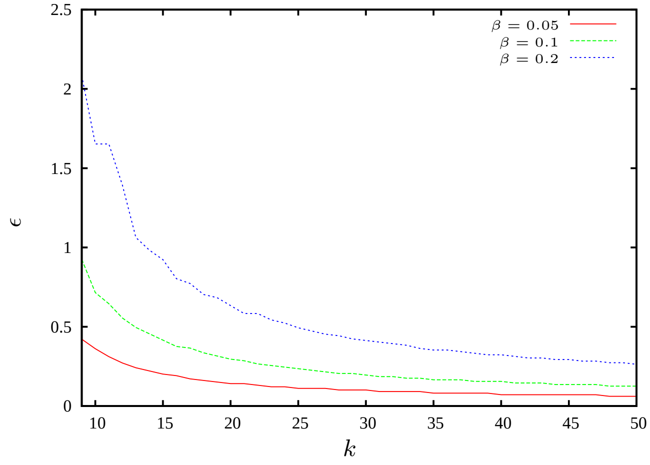

Figure 6: A graph showing the value of ϵ satisfied by a given k if $\delta \leq 10^{-6}$ with varying sampling probabilities.

The following theorem shows that ϵ-safe k-Anonymization also satisfies $(\beta, \epsilon, \delta)$-DPS.

THEOREM 6. *Any ϵ_1-safe k-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-DPS, where $\epsilon \geq -\ln(1-\beta) + \epsilon_1$, $\delta = d(k, \beta, \epsilon - \epsilon_1) = \max_{n: n \geq \lceil \frac{k}{\gamma} - 1 \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta)$, $\gamma = \frac{(e^{\epsilon - \epsilon_1} - 1 + \beta)}{e^{\epsilon - \epsilon_1}}$.*

See Appendix A.3 for the proof.

3.5 Remarks of the Result

Theorems 5 and 6 show that k-anonymization, when done safely, and when preceded by a random sampling step, can satisfy (ϵ, δ)-DP with reasonable parameters. In the literature, k-anonymization and differential privacy have been viewed as very different privacy guarantees. k-anonymization achieves weak syntactic privacy, and differential privacy provides strong semantic privacy guarantees. Our result is, to our knowledge, the first to link k-anonymization with differential privacy. This suggests that the "hiding in a crowd of k" privacy principle indeed offers some privacy guarantees when used correctly. We note that this principle is used widely in contexts other than privacy-preserving publishing of relational data, including location privacy and publishing of social network data, network packets, and other types of data.

We also observe that another way to interpret our result is that this provides a new method of satisfying (ϵ, δ)-DP. Existing methods for satisfying differential privacy include adding noise according to the global sensitivity [8], adding noise according to the smooth local sensitivity [26], and the exponential mechanism [25] which directly assigns probabilities to each possible answer in the range. Our result suggests an alternative approach: Rather than adding noises to the output, one can add a random sampling step in the beginning and prune results that are too sensitive to changes of a single individual tuple (i.e., tuples that violate k-anonymity). In other words, when the dataset is resulted from random sampling, then one can answer count queries *accurately* provided that the result is large enough. An intriguing question is whether other input perturbation techniques can be used to satisfy differential privacy as well.

4. RELATED WORK

A lot of work on privacy-preserving data publishing considers privacy notions that are weaker than differential privacy. These approaches typically assume an adversary that knows only some aspects of the dataset (background knowledge) and tries to prevent it from learning some other aspects. One can always attack such a privacy notion by changing either what the adversary already knows, or changing what the adversary tries to learn. The most prominent among these notions is k-anonymity [30, 29]. Some follow-up notions include l-diversity [23] and t-closeness [22]. In this paper, we analyze the weaknesses of k-anonymity in detail, and argue that a separation between QIDs and sensitive attributes are difficult to obtain in practice, challenging the foundation of privacy notions such as l-diversity, t-closeness, and other ones centered on attribute disclosure prevention.

The notion of differential privacy was developed in a series of works [7, 13, 3, 11, 8]. It represents a major breakthrough in privacy-preserving data analysis. In an attempt to make differential privacy more amenable to more sensitive queries, several relaxations have been developed, including (ϵ, δ)-differential privacy [7, 13, 3, 11]. Three basic general approaches to achieve differential privacy are adding Laplace noise proportional to the query's global sensitivity [8, 11], adding noise related to the smooth bound of the query's local sensitivity [26], and the exponential mechanism to select a result among all possible results [25]. A survey on these results can be found in [9]. Our approach suggests an alterative by using input perturbation rather than output perturbation to add uncertainty to the adversary's knowledge of the data.

Random sampling [1, 2] has been studied as a method for privacy preserving data mining, where privacy notions other than differential privacy were used. The relationship between sampling and differential privacy has been explored before. Chauduri and Mishra [6] studied the privacy effect of sampling, and showed a linear relationship between the sampling probability and the error probability δ. Their result suggests an approach to perform first k-anonymization and then sampling as the *last* step. We instead consider the approach of perform sampling as the *first* step and then k-anonymization. Our result suggests that the latter approach benefits much more from the sampling.

There exists some work on publishing microdata while satisfy (ϵ, δ)-DP or its variant. Machanavajjhala et al. [24] introduced a variant of (ϵ, δ)-DP called (ϵ, δ)-probabilistic differential privacy and showed that it is satisfied by a synthetic data generation method for the problem of releasing the commuting patterns of the population in the United States. This notion is stronger than (ϵ, δ)-DP. Korolova et al. [20] considered publishing search queries and clicks that achieves (ϵ, δ)-differential privacy. A similar approach for releasing query logs with differential privacy was proposed by Götz et al. [15]. These approaches apply the output perturbation technique in differential privacy to microdata publishing scenarios that can be reduced to histogram publishing at their core. Blum et al. [4] and Dwork et al. [12] considered outputing synthetic data generation that is useful for a particular class of queries. These papers do not deal with the relationship between k-anonymization and differential privacy, or between sampling and k-anonymization.

Kifer and Lin [19] developed a general framework to characterize relaxation of differential privacy. They identified two axioms for a privacy definition: Transformation Invariance and Privacy Axiom of Choice, which are satisfied by $(\beta, \epsilon, \delta)$-DPS. They did not consider the composability of these notions, which was our emphasis, as a clear understanding of the composability issues directs us what can and cannot be done with sampled dataset.

5. CONCLUSIONS

We have answered the two questions we set out in the beginning of the paper. We take the approach of starting from both k-anonymization and differential privacy and trying to meet in the middle. On the one hand, we identify weaknesses in the k-anonymity notion and existing k-anonymization methods and propose the notion of safe k-anonymization to avoid these privacy vulnerabilities. On the other hand, we try to relax differential privacy to take advantage of the adversary's uncertainty of the data. The key insight underlying our results is that random sampling can be used to bridge this gap between k-anonymization and differential privacy.

We have explored both the power and potential pitfalls to take advantage of sampling in private data analysis or publishing. Our results show that sampling, when used correctly, is a powerful tool that can greatly benefit differential privacy, as it creates uncertainty for the adversary. Sampling can increase the privacy budget and error toleration bound. Sampling also enables the usage of algorithms such as safe k-anonymization; however, this usage requires fresh sampling that is not used to answer any other query. An intriguing open question is whether there exist approaches other than sampling that can create uncertainty for the adversary, that can tolerate answering ϵ-DP queries.

6. REFERENCES

[1] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving olap. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 251–262, 2005.

[2] S. Agrawal and J. R. Haritsa. A framework for high-accuracy privacy-preserving mining. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 193–204, 2005.

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, New York, NY, USA, 2005. ACM.

[4] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[5] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *Proceedings of the 12th international conference on Database systems for advanced applications*, DASFAA'07, pages 188–200, 2007.

[6] K. Chaudhuri and N. Mishra. When random sampling preserves privacy. In *CRYPTO*, pages 198–213, 2006.

[7] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, New York, NY, USA, 2003. ACM.

[8] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[9] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In S. Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[12] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[13] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *In CRYPTO*, pages 528–544. Springer, 2004.

[14] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7:1–18, January 2008.

[15] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Privacy in search logs. *CoRR*, abs/0904.0682, 2009.

[16] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, page ss, 2009.

[17] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 320–326, 1992.

[18] S. P. Kasiviswanathan and A. Smith. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*, abs/0803.3946, 2008.

[19] D. Kifer and B.-R. Lin. Towards an axiomatization of statistical privacy and utility. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems of data*, PODS '10, pages 147–158, New York, NY, USA, 2010. ACM.

[20] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 171–180, 2009.

[21] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, page 25, 2006.

[22] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.

[23] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, page 24, 2006.

[24] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, pages 277–286, 2008.

[25] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[26] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007.

[27] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13:1010–1027, November 2001.

[28] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[29] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[30] L. Sweeney. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

APPENDIX

A. PROOFS

This appendix includes proofs not included in the main body.

A.1 Proof of Theorem 1

Theorem 1. *Given an algorithm $\mathcal{A}$ that satisfies $(\beta_1, \epsilon_1, \delta_1)$-DPS, $\mathcal{A}$ also satisfies $(\beta_2, \epsilon_2, \delta_2)$-DPS for any $\beta_2 < \beta_1$, where*
$\epsilon_2 = \ln\left(1 + \left(\frac{\beta_2}{\beta_1}(e^{\epsilon}_1 - 1)\right)\right)$ *and* $\delta_2 = \frac{\beta_2}{\beta_1}\delta_1$.

Proof. We need to show that the algorithm $\mathcal{A}^{\beta_2}$ satisfies (ϵ_2, δ_2)-DP. Let $\beta = \frac{\beta_2}{\beta_1}$. The algorithm $\mathcal{A}^{\beta_2}$ can be viewed as first sampling with probability β, then followed by applying the algorithm $\mathcal{A}^{\beta_1}$, which satisfies (ϵ_1, δ_1)-DP.

We use $\Lambda_\beta(D)$ to denote the process of sampling from D with sampling rate β. Any pair D, D' can be viewed as D and D_{-t}, where D_{-t} denotes the dataset resulted from removing one copy of t from D. For any O, let

$$Z = \Pr[\mathcal{A}^{\beta_2}(D) \in O], \text{ and } X = \Pr[\mathcal{A}^{\beta_2}(D_{-t}) \in O],$$

we want to show that

$$(Z \leq e^{\epsilon_2} X + \delta_2) \wedge (X \leq e^{\epsilon_2} Z + \delta_2).$$

We have

$$Z = \sum_{T \subseteq D} \Pr[\Lambda_\beta(D) = T] \Pr[\mathcal{A}^{\beta_1}(T) \in O],$$

$$X = \sum_{T \subseteq D_{-t}} \Pr[\Lambda_\beta(D_{-t}) = T] \Pr[\mathcal{A}^{\beta_1}(T) \in O].$$

To analyze Z, we note that all the T's that resulted from sampling from D with probability β can be divided into those in which t is not sampled, and those in which t is sampled. For a T in the former case, we have

$$\begin{aligned} \Pr[\Lambda_\beta(D) = T] &= (1-\beta) \Pr[\Lambda_\beta(D) = T | t \text{ not sampled in } T] \\ &= (1-\beta) \Pr[\Lambda_\beta(D_{-t}) = T] \end{aligned}$$

For a T in the latter case, we have

$$\begin{aligned} \Pr[\Lambda_\beta(D) = T] &= \beta \Pr[\Lambda_\beta(D) = T | t \text{ sampled in } T] \\ &= \beta \Pr[\Lambda_\beta(D_{-t}) = T_{-t}]. \end{aligned}$$

Hence we have

$$\begin{aligned} Z = \quad & \textstyle\sum_{T \subseteq D_{-t}} (1-\beta) \Pr[\Lambda_\beta(D_{-t}) = T] \Pr[\mathcal{A}^{\beta_1}(T) \in O] \\ & + \textstyle\sum_{T_{-t} \subseteq D_{-t}} \beta \Pr[\Lambda_\beta(D_{-t}) = T_{-t}] \Pr[\mathcal{A}^{\beta_1}(T_{-t}) \in O] \end{aligned}$$

Let

$$Y = \sum_{T' \subseteq D_{-t}} \Pr[\Lambda_\beta(D_{-t}) = T'] \Pr[\mathcal{A}^{\beta_1}(T'_{+t}) \in O],$$

then we have $Z = (1-\beta)X + \beta Y$.

That $\mathcal{A}$ satisfies $(\beta_1, \epsilon_1, \delta_1)$-DPS means that for each T, O

$$\Pr[\mathcal{A}^{\beta_1}(T_{+t}) \in O] \leq e^{\epsilon_1} \Pr[\mathcal{A}^{\beta_1}(T) \in O] + \delta_1$$

Hence we have

$$\begin{aligned} Y \quad & \leq \textstyle\sum_{T' \subseteq D_{-t}} \Pr[\Lambda_\beta(D_{-t}) = T'] \left(e^{\epsilon_1} \Pr[\mathcal{A}^{\beta_1}(T' \in O] + \delta_1 \right), \\ & = e^{\epsilon_1} \textstyle\sum_{T' \subseteq D_{-t}} \Pr[\Lambda_\beta(D_{-t}) = T'] \Pr[\mathcal{A}^{\beta_1}(T') \in O] \\ & \quad + \delta_1 \textstyle\sum_{T' \subseteq D_{-t}} \Pr[\Lambda_\beta(D_{-t}) = T'] \\ & = e^{\epsilon_1} X + \delta_1. \end{aligned}$$

Hence we have

$$\begin{aligned} Z &= (1-\beta)X + \beta Y \\ &\leq (1-\beta)X + \beta(e^{\epsilon_1} X + \delta_1) \\ &\leq (1-\beta+\beta e^{\epsilon_1})X + \beta\delta_1. \\ &= e^{\epsilon_2} X + \delta_2. \end{aligned}$$

To show that $X \leq e^{\epsilon_2} Z + \delta_2$, we observe that $\mathcal{A}$ satisfies $(\beta_1, \epsilon_1, \delta_1)$-DPS means that

$$X \leq e^{\epsilon_1} Y + \delta_1, \text{ and hence}$$

$$Z = (1-\beta)X + \beta Y \geq (1-\beta)X + \beta e^{-\epsilon_1}(X - \delta_1),$$

$$\text{and } X \leq \frac{1}{1-\beta+\beta e^{-\epsilon_1}} Z + \frac{\beta e^{-\epsilon_1}}{1-\beta+e\beta^{-\epsilon_1}} \delta_1$$

We now show that

$$\frac{1}{1-\beta+\beta e^{-\epsilon_1}} \leq e^{\epsilon_2 = \ln\left(1+\left(\frac{\beta_2}{\beta_1}(e^{\epsilon_1}-1)\right)\right)} = 1+\beta(e^{\epsilon_1}-1) = e^{\epsilon_2}.$$

$$\begin{aligned} & \tfrac{1}{1-\beta(1-e^{-\epsilon_1})} \leq 1 + \beta(e^{\epsilon_1} - 1) \\ \Leftrightarrow \quad & 0 \leq (1 + \beta(e^{\epsilon_1} - 1))(1 - \beta(1 - e^{-\epsilon_1})) - 1 \\ \Leftrightarrow \quad & 0 \leq (e^{\epsilon_1} + e^{-\epsilon_1} - 2)(\beta - \beta^2). \end{aligned}$$

Hence

$$\frac{\beta e^{-\epsilon_1}}{1-\beta+e\beta^{-\epsilon_1}} \delta_1 \leq \beta e^{-\epsilon_1} e^{\epsilon_2} \delta_1 \leq \beta\delta_1 = \delta_2.$$

□

A.2 Proof of Theorem 5

Theorem 5: *Any strongly-safe k-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-DPS for any $0 < \beta < 1$, $\epsilon \geq -\ln(1-\beta)$, and $\delta = d(k, \beta, \epsilon)$, where the function d is defined as*

$$d(k, \beta, \epsilon) \quad = \max_{n: n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta),$$

where $\gamma = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon}$.

PROOF. Let $\mathcal{A}$ denote the algorithm, and g be the data-independent generalization procedure in the algorithm. For any dataset D, any tuple $t \in D$, and for any output S. For any $\epsilon \geq -\ln(1-\beta)$, we show that the probability by which

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} \leq e^{\epsilon} \qquad (2)$$

is violated is δ. Note that this is a stronger version of (ϵ, δ)-DP than the one in Definition 2. See [18] for relationship between the two.

Let n be the number of t' in D such that $g(t') = g(t)$. Let j be the number of times that $g(t)$ appears in S. Note that as the only difference between D and D_{-t} is that D has one extra copy of t, we have.

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t})) = S]} = \frac{\Pr[\mathcal{A}(D) \text{ has } j \text{ copies of g}(t)]}{\Pr[\mathcal{A}(D_{-t}) \text{ has } j \text{ copies of g}(t)]}$$

Because any tuple that appears less than k times is suppressed, either $j \geq k$, or $j = 0$. When $j = 0$, we have

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} = \frac{F(k-1; n, \beta)}{F(k-1; n-1, \beta)} = \frac{\sum_{i=0}^{k-1} f(i; n, \beta)}{\sum_{i=0}^{k-1} f(i; n-1, \beta)}.$$

Because $F(k-1; n, \beta)$ is always less than $F(k-1; n-1, \beta)$;[3] hence $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t})=S]} < e^\epsilon$. Furthermore, we note that $\forall i \in [0..k-1]$, $\frac{f(i;n,\beta)}{f(i;n-1,\beta)} = \frac{n(1-\beta)}{n-i} \geq (1-\beta)$. Hence $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t}))=S]} \geq (1-\beta)$. Because $\epsilon \geq -\ln(1-\beta)$, we have $e^{-\epsilon} \leq 1-\beta$; hence under the case when $j = 0$, inequality (2) is satisfied.

When $j \geq k$, we have

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}5)) = S]} = \frac{f(j; n, \beta)}{f(j; n-1, \beta)} = \begin{cases} \frac{n(1-\beta)}{n-j} & n \geq j \\ 1 & n < j. \end{cases}$$

The choice of n can be arbitrary because it is determined by the choice of D. The value of j is determined by the choice of S. For some values of j, inequality (2) is violated. We want to compute the probabilities of these bad j's occurring. From the above, we know when $j > n$, the outcome is good. We now consider the bad outcomes when $j \leq n$.

9[3]Let X_i's be random variables that take the value 1 with probability β, and 0 with probability $1-\beta$. $F(k-1; n-1, \beta)$ is the probability that the sum of $n-1$ such X's $\leq k-1$, and $F(k-1; n, \beta)$ is the probability that the sum of n such X's is $\leq k-1$.

Note that because $\epsilon \geq -\ln(1-\beta)$, we have $-\epsilon \leq \ln(1-\beta)$, and

$$\frac{n(1-\beta)}{n-j} > 1-\beta \geq e^{-\epsilon}.$$

Hence we only need to consider what j's make $\frac{n(1-\beta)}{n-j} > e^{\epsilon}$. This occurs when $j > \frac{(e^{\epsilon}-1+\beta)n}{e^{\epsilon}}$. Let $\gamma = \frac{(e^{\epsilon}-1+\beta)}{e^{\epsilon}}$, then this occurs when $j > \gamma n$.

So far our analysis has shown that 5a bad outcome S for an input D would satisfy the condition $j \geq k$ and $n \geq j > \gamma n$. Now we need to compute the probability that $\mathcal{A}(D)$ gives a bad outcome, and the probability that $\mathcal{A}(D_{-t})$ gives a bad outcome. The former is given below:

$$\max_{n} \sum_{j:(j \geq k \wedge j > \gamma n)}^{n} f(j; n, \beta) \tag{3}$$

And the latter is

$$\max_{n} \sum_{j:(j \geq k \wedge j > \gamma n)}^{n-1} f(j; n-1, \beta).$$

As the latter is smaller than the former, we only need to bound the former.

Let $n_m = \left\lceil \frac{k}{\gamma} - 1 \right\rceil$, we now show that when $n \leq n_m$, $\sum_{j:(j \geq k \wedge j > \gamma n)}^{n} f(j; n, \beta)$ increases when n increases. Note that the choice of n_m satisfies the condition that $\gamma n_m < k$ and $\gamma(n_m + 1) \geq k$. Observe that when $n \leq n_m$, the condition $(j \geq k \wedge j > \gamma n)$ becomes $j \geq k$. The function $\sum_{j:j \geq k}^{n} f(j; n, \beta)$ is monotonically increasing with respect to n.

When $n \geq n_m$, the condition $(j \geq k \wedge j > \gamma n)$ becomes $j > \gamma n$. (In fact, when $n = n_m + 1$, the smallest j to satisfy $j > \gamma n$ is $k+1$.) Hence the error probability is bounded by

$$\delta = d(k, \beta, \epsilon) = \max_{n: n \geq \lceil \frac{k}{\gamma} - 1 \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta), \text{ where } \gamma = \frac{(e^{\epsilon}-1+\beta)}{e^{\epsilon}}.$$

□

A.3 Proof of Theorem 6

Theorem 6: *Any ϵ_1-safe k-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-DPS, where $\epsilon \geq -\ln(1-\beta) + \epsilon_1$, $\delta = d(k, \beta, \epsilon - \epsilon_1) = \max_{n: n \geq \lceil \frac{k}{\gamma} - 1 \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta)$, $\gamma = \frac{(e^{\epsilon - \epsilon_1} - 1 + \beta)}{e^{\epsilon - \epsilon_1}}$.*

PROOF. Let $\mathcal{A}$ denote the ϵ_1-safe k-anonymization algorithm. Here, we want to show that for any $\epsilon \geq -\ln(1-\beta) + \epsilon_1$, D, $t \in D$ and S,

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} \leq e^{\epsilon} \tag{4}$$

is valid for probability at least $1 - \delta$. Let Λ_β denote the process of binomial sampling the dataset D with probability β. And let G denote the set of all the possible outputs of $\mathcal{A}$'s subroutine $\mathcal{A}_m$. By definition, its subroutine $\mathcal{A}_m$ satisfies ϵ_1-differential privacy,

$$e^{-\epsilon_1} \leq \frac{\Pr[\mathcal{A}_m(D) = g]}{\Pr[\mathcal{A}_m(D_{-t}) = g]} \leq e^{\epsilon_1}.$$

And, according to the proof of Theorem 6, for a fixed $g \in G$, the ratio $r(g) = \frac{\Pr[g(\Lambda_\beta(D))=S]}{\Pr[g(\Lambda_\beta(D_{-t}))=S]}$ equals

$$r(g) = \begin{cases} \frac{\sum_{i=0}^{k-1} f(i;n,\beta)}{\sum_{i=0}^{k-1} f(i;n-1,\beta)} & \text{if } j = 0; \\ \frac{n(1-\beta)}{n-j} & \text{if } k \leq j \leq n \end{cases}$$

where j is the number of copies of $g(t)$ in the output dataset S. So, the differential privacy ratio (4) can be upper bounded,

$$\begin{aligned} & \frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t})=S]} \\ = & \frac{\sum_{g \in G} \Pr[\mathcal{A}_m(D)=g] \cdot \Pr[g(\Lambda_\beta(D))=S]}{\sum_{g \in G} \Pr[\mathcal{A}_m(D_{-t})=g] \cdot \Pr[g(\Lambda_\beta(D_{-t}))=S]} \\ \leq & \frac{e^{\epsilon 1} \sum_{g \in G} \Pr[\mathcal{A}_m(D_{-t})=g] \cdot \Pr[g(\Lambda_\beta(D))=S]}{\sum_{g \in G} \Pr[\mathcal{A}_m(D_{-t})=g] \cdot \Pr[g(\Lambda_\beta(D_{-t}))=S]} \\ \leq & \frac{e^{\epsilon 1} r(g) \sum_{g \in G} \Pr[\mathcal{A}_m(D_{-t})=g] \cdot \Pr[g(\Lambda_\beta(D_{-t}))=S]}{\sum_{g \in G} \Pr[\mathcal{A}_m(D_{-t})=g] \cdot \Pr[g(\Lambda_\beta(D_{-t}))=S]} \\ = & e^{\epsilon 1} r(g). \end{aligned}$$

The lower bound can be obtained in a similar way. So,

$$e^{-\epsilon_1} r(g) \leq \frac{\Pr[\mathcal{A}^\beta(D) = S]}{\Pr[\mathcal{A}^\beta(D_{-t}) = S]} \leq e^{\epsilon_1} r(g).$$

By the proof of Theorem 6, the ratio $r(g)$ is bounded by $e^{-(\epsilon-\epsilon_1)} \leq r(g) \leq e^{(\epsilon-\epsilon_1)}$. The probability that it is violated is the probability that inequality (4) is violated. In the $j = 0$ case, $e^{-(\epsilon-\epsilon_1)} \leq \frac{\sum_{i=0}^{k-1} f(i;n,\beta)}{\sum_{i=0}^{k-1} f(i;n-1,\beta)} \leq e^{(\epsilon-\epsilon_1)}$, since $\epsilon \geq -\ln(1-\beta) + \epsilon_1$. And for the $k \leq j \leq n$ case, $\frac{n(1-\beta)}{n-j} > (1-\beta) \geq e^{-\epsilon+\epsilon_1}$. And only when $\frac{n(1-\beta)}{n-j} > e^{\epsilon-\epsilon_1}$ $\left(j > \frac{n(e^{\epsilon-\epsilon_1}-1+\beta)}{e^{\epsilon-\epsilon_1}}\right)$, inequality (4) is violated. Let $\gamma = \frac{(e^{\epsilon-\epsilon_1}-1+\beta)}{e^{\epsilon-\epsilon_1}}$. The error probability δ is

$$\delta = d(k, \beta, \epsilon - \epsilon_1) = \max_{n: n \geq \lceil \frac{k}{\gamma} - 1 \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta),$$

where $\gamma = \frac{(e^{\epsilon-\epsilon_1}-1+\beta)}{e^{\epsilon-\epsilon_1}}$. □